Fruit Detection in an Apple Orchard Using a Mobile Terrestrial Laser Scanner

Jordi Gené-Mola[a], Eduard Gregorio[a,*], Javier Guevara[b], Fernando Auat[b], Ricardo Sanz-Cortiella[a], Alexandre Escolà[a], Jordi Llorens[a], Josep-Ramon Morros[c], Javier Ruiz-Hidalgo[c], Verónica Vilaplana[c], Joan R. Rosell-Polo[a]

[a]*Research Group in AgroICT & Precision Agriculture, Department of Agricultural and Forest Engineering, Universitat de Lleida (UdL) – Agrotecnio Center, Lleida, Catalonia, Spain.*

[b]*Department of Electronic Engineering, Universidad Técnica Federico Santa María, Valparaíso, Chile.*

[c]*Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, Barcelona, Catalonia, Spain.*

Abstract

The development of reliable fruit detection and localization systems provides an opportunity to improve the crop value and management by limiting fruit spoilage and optimized harvesting practices. Most proposed systems for fruit detection are based on RGB cameras and thus are affected by intrinsic constraints, such as variable lighting conditions and camera calibration. This work presents a new technique that uses a mobile terrestrial laser scanner (MTLS) to detect and localise Fuji apples. An experimental test focused on Fuji apple trees (*Malus domestica* Borkh. cv. Fuji) was carried out. A 3D point cloud of the scene was generated using an MTLS composed of a Velodyne VLP-16 LiDAR sensor synchronized with an RTK-GNSS satellite navigation receiver. A reflectance analysis of tree elements was performed, obtaining mean apparent reflectance values of 28.9%, 29.1%, and 44.3% for leaves, branches and trunks, and apples, respectively. These results suggest that the apparent reflectance parameter (at 905 nm wavelength) can be useful to detect apples in the tree. For that purpose, a four-step fruit detection algorithm was developed. By applying this algorithm, a localization success of 87.5%, an identification success of 82.4%, and an F1-score of 0.858 were obtained in relation to the total amount of fruits. These detection rates are similar to those obtained by RGB-based systems, but with the additional advantages of providing direct 3D fruit location information, which is not affected by sunlight variations. From the experimental results, it can be concluded that LiDAR-based technology and, particularly, its reflectance information, has potential for remote apple detection and 3D location.

Keywords: LiDAR; Mobile Terrestrial Laser Scanning; Fruit detection; Agricultural robotics; Fruit reflectance

Nomenclature

FDR_{ID}	False detection rate identification
FDR_L	False detection rate localization
FoV	Field of View *[º]*
FP_{ID}	False positive identification
FP_L	False positive localization

GT_{field}	Number of fruits manually-counted in field
GT_{labels}	Number of fruits labelled
IoD_i	Intersection over detection
K	Number of fruits in a cluster
MTLS	Mobile Terrestrial Laser Scanner
n	Number of clusters that detect the same fruit
N_m	Fruit multi-detections (n-1)
P	Number of points of a cluster
P_{kj}	Number of points threshold used to find clusters with j apples
R	Apparent reflectance *[%]*
R_{th}	Reflectance threshold *[%]*
RTK-GNSS	Real-Time Kinematics Global Navigation Satellite System
$\bar{R}$	Mean apparent reflectance of the points of a cluster *[%]*
$\bar{R}_{FP}$	Mean apparent reflectance threshold used to find false positive clusters *[%]*
$\bar{R}_{kj}$	Mean apparent reflectance threshold used to find clusters with j apples *[%]*
$Success_{ID}$	Identification success (recall)
$Success_L$	Localization success
SVD	Singular Value Decomposition
TOF	Time of flight
TP_{ID}	True positive identification
TP_L	True positive localization
V	Volume of a cluster $[m^3]$
V_{FP}	Volume threshold used to find false positive clusters $[m^3]$
V_{kj}	Volume threshold used to find clusters with j apples $[m^3]$
[x, y, z]	3D point with UTM coordinates *[m]*
α	Sparse outlier removal tuning parameter
λ_{in}	Normalized principal value i
λ_i	Principal value i of a cluster
Ψ	Geometric parameter
Ψ_{FP}	Geometric parameter value used to find false positive clusters
Ψ_{kj}	Geometric parameter value used to find clusters with j apples

1. Introduction

Fruticulture is under constant pressure to increase fruit production and quality, as demanded by a growing world population. To this end, farmers need to find new ways to improve fruit productivity and, at the same time, reduce economic and environmental costs (Siegel *et al.*, 2014). Agricultural robotics takes advantage of new technologies to respond to this challenge (Bac *et al*., 2014; Bechar and Vigneault, 2017, 2016; Gongal *et al*., 2015; Y. Zhao *et al*., 2016). The use of robotics in agricultural fields and orchards is increasing, particularly in tasks related to guidance (seeding or harvesting), detection (weed monitoring and control, extraction of biological features), and mapping (Auat Cheein *et al*., 2017; Auat Cheein and Carelli, 2013; Foglia and Reina, 2006). In general, the development of intelligent robots interacting with agricultural fields increases the accuracy of tasks and reduces the consumption of resources without decreasing yield, making it a reasonable option for repeatable tasks (Cariou *et al*., 2009; Foglia and Reina, 2006; Zhang and Pierce, 2016).

Fruit detection and localization are complex tasks that can be handled by agricultural robotics, with applications related to yield prediction, yield mapping, and automated harvesting. Nowadays, yield prediction is done by manual counting of selected sample trees, leading to inaccurate predictions due to the high variability in orchards (Payne *et al*., 2014; Stein *et al*., 2016). Crop monitoring using new technologies could provide more accurate and efficient predictions (Bechar and Vigneault, 2017, 2016). Another application of fruit detection is yield mapping. The fruit load of an orchard is influenced by in-field spatial variability (due to soil type variations), fertility, and water content, among other factors. In precision agriculture, yield mapping helps to determine the reasons for and find solutions to cope with this variability (Kurtulmus *et al*., 2014). Finally, fruit localization is the basis for future automated harvesting. Manual picking is a bottleneck in fruit production management, because it requires lots of resources in the context of decreasing farming labour force. In addition, hand harvesting exposes farmers to awkward postures on ladders and platforms with heavy loads, making manual harvesting dangerous and inefficient (De-An *et al*., 2011; Gongal *et al*., 2015).

The detection of fruits can involve many fruit properties of different complexity, from the simplest, such as the presence/absence of a fruit, to properties that are more challenging to measure, including size, volume, diameter, maturation stage, sugar, and other substance contents, defects and disease/pest affectation, etc. There are multiple technologies available for fruit detection and localization, each with its advantages and disadvantages (Gongal *et al*., 2015). All approaches have to solve problems derived from occlusions (Stein *et al*., 2016; Wachs *et al*., 2010), clustering (Gong *et al*., 2013; Xiang *et al*., 2014), and variable lighting conditions (Gongal *et al*., 2016; C. Zhao *et al*., 2016).

The most commonly used sensors are RGB cameras (Linker, 2017; Maldonado and Barbosa, 2016; C. Zhao *et al*., 2016). These are affordable sensors, which allow fruits to be distinguished from other elements by colour (Linker *et al*.,

2012; Liu *et al*., 2016), geometric shape (Barnea *et al*., 2016; Lak *et al*., 2010), texture (Chaivivatrakul and Dailey, 2014; Qureshi *et al*., 2017), or by using machine learning techniques like, e.g., deep neural networks (Bargoti and Underwood, 2017). The two main drawbacks to RGB cameras are their sensitivity to lighting conditions and the fact that they only provide 2D information (unless using stereoscopic techniques). Other, more expensive, cameras include thermal cameras (Bulanon *et al*., 2009, 2008; Stajnko *et al*., 2004; Wachs *et al*., 2010), multispectral cameras (Sa *et al*., 2016; Zhang *et al*., 2015), and hyperspectral cameras (Okamoto and Lee, 2009; Safren *et al*., 2007). The former allows fruits to be distinguished from the background through their temperature, while the latter detect fruits from their reflectance at different wavelengths. Like RGB cameras, thermal, multispectral, and hyperspectral cameras do not provide 3D information, unless a stereoscopic approach is implemented.

There are several solutions to obtain three-dimensional information. One of them is based on using two (stereovision) or more cameras (Font *et al*., 2014; Si *et al*., 2015; Xiang *et al*., 2014). By applying triangulation techniques, it is possible to obtain the depth of each pixel and reconstruct the 3D structure. The major advantage of this technique is that it allows us to obtain accurate 3D models with RGB information, while the main disadvantages are that 3D model generation is computationally expensive and the performance is affected by lighting conditions. Another more recent technique is the use of laser range finders and LiDAR-based (Light Detection and Ranging) systems. These are more expensive sensors that generally operate under the principle of time-of-flight (TOF) (Wehr and Lohr, 1999). This type of sensor typically also provides the amount of energy backscattered from the impacted object. Very few studies have used LiDAR-based systems in fruit detection and, to the best of the authors' knowledge, none of them have been tested in a real orchard environment. For example, Jiménez *et al*. (2000, 1999) developed a vision system based on a laser range-finder, with the aim of detecting spherical objects in non-structured environments. They report good detection performances, although the tests were carried out on a limited number of oranges suspended from an artificial tree. Finally, another technology derived from photogrammetry and LiDAR, and also used in fruit growing, are the RGB-D (depth) cameras, where each pixel of the image contains colour and depth data, generating 3D colour images (Barnea *et al*., 2016; Nguyen *et al*., 2016; Rosell-Polo *et al*., 2017, 2015). These systems are based on the simultaneous combination of RGB cameras and depth sensors based on laser light (either through structured laser light or TOF flash-type LiDAR-based systems).

This work presents a proof of concept of using LiDAR in detecting Fuji apples in producing orchard trees. The methodology is founded on the fact that apples have higher apparent reflectance than leaves and trunks at 905 nm laser

wavelength. The main contributions of this paper are: (1) analysis of apple reflectivity on 3D point clouds from LiDAR sensors; (2) development of an apple detection and localization algorithm based on three stages (point cloud segmentation; fruit separation, and false positive removal); and (3) experimental validation of the proposed technique on a real Fuji apple orchard. The principal advantage of this technique over previously published efforts would be its capacity to provide direct 3D fruit localization information without being affected by illumination conditions. The paper is structured as follows. Section 2 presents the experimental data set, the point cloud generation procedure, the reflectance analysis, and the developed apple detection algorithm. Section 3 shows the results of the first experimental tests performed on three Fuji apple trees of a commercial orchard. Finally, the conclusions are presented in Section 4.

2. Materials and methods

2.1. Experimental set up

A fruit detection experiment was carried out on September 28th of 2017 in Tarassó farm, a commercial apple orchard located in Agramunt, Catalonia, Spain (E: 336,297 m; N: 4,623,494 m; 312 m a.s.l., UTM 31T - ETRS89). The trials were carried out in an 8-year-old Fuji apple orchard (*Malus domestica* Borkh. cv. Fuji), trained in a tall spindle system with a maximum tree height of 3.75 m. The three analysed trees were at BBCH (Biologische Bundesanstalt, Bundessortenamt und CHemische Industrie) growth stage 85 (Meier *et al.*, 2001), three weeks before harvesting.

The measurement equipment consisted of a mobile Terrestrial Laser Scanner (MTLS), comprised of a LiDAR sensor and a real-time kinematics global navigation satellite system (RTK-GNSS), connected to a rugged laptop suitable for working in field conditions. The LiDAR sensor used was a Puck VLP-16 (Velodyne LIDAR Inc., San José, CA, USA), which generates a 3D point cloud (x-y-z positions) of the scanned scene, as well as calibrated apparent reflectance (R) of each point in the 3D point cloud. This calibration was carried out by sensor manufacturer using a set of calibration targets, and implies a conversion of the backscattered range-corrected intensity (digital numbers) into apparent reflectance values independently of laser power and distance (Velodyne, 2016). Note that the measured apparent reflectance (hereinafter referred to as reflectance) is an approximation of the actual hemispherical reflectance, considering that the measured objects are Lambertian (diffuse reflectors), and not considering the incidence angle (Kaasalainen et al., 2011; Kukko et al., 2008; Ray, 1994). The VLP-16 sensor emits 16 laser beams (905 nm wavelength) with a horizontal angular resolution of 2º (30º horizontal FoV) when mounted on a vertical plane as shown in **Fig. 1**. Although the vertical FoV can be set up to

360º, in this experiment it was set to 150º, since only one row of trees was scanned. The scanning frequency rate was set to 10 Hz, corresponding to a vertical angular resolution of 0.2º, so that a maximum of 12,000 points were obtained from each scan (acquisition speed of 120,000 points/second). Even though this sensor has a range of 100 m, points further than 4 m where not considered for 3D point cloud generation, thus only the tree row of interest was modelled. The acquisition of Coordinated Universal Time (UTC) of each point was obtained via a GPS 18x LVC receiver (Garmin International Inc., Olathe, KS, USA), connected to the VLP-16 sensor. The RTK-GNSS system used was the GPS1200+ (Leica Geosystems AG, Heerbrugg, Swizeland), which provides absolute coordinates and UTC time (synchronized with the LiDAR) with a frequency of 20 Hz and a precision of approx. 20 mm.

Fig. 1. Should be placed here

As shown in **Fig. 1**, the MTLS measurement system was mounted on the rear of an air-assisted sprayer by means of an aluminium structure. The sprayer was pulled at low gear by a farm tractor equipped with an electronic speedometer. The GNSS rover receiver antenna was installed on top of the mast, at a height of 3.5 m. The LiDAR sensor was mounted vertically (**Fig. 1**) and placed at a height of 1.8 m, that is about half the maximum height of studied trees. This position was selected to have similar detection performance along the tree height. The field test was performed by moving the MTLS along a rectilinear trajectory parallel to the tree row axis, at a distance of 2.4 m. Due to the fact that the system did not include an inertial measurement unit (IMU), moving the MTLS along a linear trajectory was important to improve the point cloud consistency. The forward speed was 0.125 m s^{-1}, corresponding to a resolution of 12.5 mm between consecutive scans (~53,600 points m^{-2} in a vertical plane at the distance of 2.4m). The tree row was scanned from both sides in order to obtain a complete 3D model.

2.2. 3D point cloud model

A rigid transformation was performed by applying a rotation and translation matrix to each point, in order to build the point cloud with absolute coordinates. The translation matrix was built using the Universal Transverse Mercator (UTM) coordinates of the RTK-GNSS system, by considering the relative distance between the optical centre of the LiDAR sensor and the GNSS receiver. The rotation matrix depends on the orientation of the MTLS at each time instant and was obtained by the forward direction computed from the measurements of the RTK-GNSS receiver. Given that the trials were

performed with a short rectilinear trajectory, the tilt of the platform can be ignored, assuming a constant orientation along the path. An illustration of the 3D point cloud models generated is shown in **Fig. 2**.

The resulting 3D point cloud was manually labelled in order to generate ground truth of the apples locations. This enables a study of the features that characterize the apples, as well as the possibility to evaluate the performance of the developed apple detection techniques. The annotation was carried out using the software CloudCompare (Cloud Compare [GPL software] v2.9 Omnia), placing 3D rectangular bounding boxes on each apple, as can be seen in the third tree of **Fig. 2**. This annotation was supported by additional RGB images to localise the apples in the 3D point cloud. The actual number of apples counted in field (ground truth field or *GT_field*) were 139 in tree 1, 145 in tree 2, and 139 in tree 3, of which 133, 138 and 134, respectively, could be labelled in the 3D point cloud (ground truth labels or *GT_labels*) due to occlusions or in field counting errors. Trees 1 and 2 were used as the training dataset to select and tune the algorithm parameters, while tree 3 was used as the test dataset to evaluate the performance of the developed algorithm. From the labelled scene and the reflectance data extracted from the LiDAR for each point, a reflectance study of the different elements of the tree was carried out (Section 3.1).

Fig. 2. Should be placed here

2.3. Apple detection algorithm

As shown in **Fig. 3**, the algorithm proposed in this paper is structured as follows: 1) Point cloud segmentation; 2) fruit separation; and 3) false positive removal. The segmentation is based on the reflectance of measured elements and aims at removing points corresponding to leaves, branches, and the trunk, and grouping the remaining points -likely to be an apple- in clusters. The fruit separation uses features of clusters in order to identify and split those that contain more than one apple. False positive removal is based on the geometry and reflectance of the clusters. All data processing was implemented in MATLAB (R2018a, Math Works Inc., Natick, Massachusetts, USA). The different implemented steps are detailed below.

Fig. 3. Should be placed here

2.3.1. Point cloud segmentation

The objective of this step is to segment the 3D point cloud and obtain a set of clusters with points that could be apple candidates. Since some groups of apples could be touching, the clusters obtained in this first step could contain one or more apples. The 3D model acquired with the MTLS consists of a set of 3D points with UTM coordinates and their reflectance

197 *[x, y, z, R]*. The reflectance analysis (Section 3.1) shows that apple reflectance at the 905 nm laser wavelength is higher
198 than that for leaves and the trunk and, therefore, this parameter is used for apple detection. To remove the points that do not
199 correspond to apples, a threshold, R_{th}, is applied. This is followed by Sparse Outlier Removal (Rusu *et al*., 2008) to reduce
200 the noise; this approach removes the points which fall outside $\mu + \alpha \cdot \sigma$, with μ and σ being the mean and standard deviation,
201 respectively, of the k nearest neighbour distances, while α is a tuning parameter. The point cloud segmentation ends with a
202 connected components labelling using a density-based scan algorithm, DBSCAN (Ester *et al*., 1996), which clusters points
203 that have more than minPts points closer than a distance, ε. Outlier Removal is first applied to delete noisy points that
204 otherwise would connect clusters from different apples. All the parameters used in this step were selected through a
205 hyperparameter optimization procedure, using the training data set to search for the combination of parameters that best
206 suits our data. In this search, we found that the results were stable against small variations in the different parameter values,
207 except the reflectance threshold, the behaviour of which is shown in Section 3.1, **Fig. 7**. The parameter values used are
208 detailed in Appendix A.

209 *2.3.2. Apple separation*

210 If apples are properly separated, the results obtained in the previous step would consist of a set of clusters of one apple
211 in each. Nevertheless, it was found that groups of apples touching will result in clusters of more than one apple. The aim of
212 this second step is to identify the clusters containing more than one apple and split them into sub-clusters, each containing
213 one apple. First, the number of apples, *K*, that make up a cluster has to be predicted, and then the cluster is split using the
214 K-means algorithm. This clustering method aims to partition the 3D points into K sub-clusters in which each 3D point
215 belongs to the sub-cluster with nearest mean (Jain, 2010).

216 To predict the number of apples contained in each cluster (the K number used in the K-means algorithm), three different
217 methods were tested. The first one is inspired by a template matching technique (Brunelli, 2009). The second method
218 applies a decision tree, based on cluster features such as volume, density of points, reflectance, and shape. Finally, the third
219 method is a combination of the previous two approaches. These methods are explained in more detail below.

220 *Method 1*

221 The first approach projects the 3D point clouds of each cluster (Fig. 4.b) onto a 2D plane, obtaining an image of the
222 cluster with reflectance data at a resolution of 4x4 mm per pixel (Fig. 4.c). The cluster image then is convolved with a

Gaussian filter of size 20x20 pixels and standard deviation of 3.5. These parameters correspond to the measured fruit size, so that the dimension of this filter is 80x80 mm, similar to the mean size of the tested apples. Since the apples have an approximately spherical shape, when the cluster image is convolved with a Gaussian filter, the local maxima of the obtained image correspond to the centres of the apples (Fig. 4.d). The value K, to be used in the K-means algorithm, corresponds to the number of local maxima found in the convolved image (Fig. 4.e).

The result of this method could vary with the 2D projection plane used (e.g., the projection may produce occlusions). The technique is applied in four different planes to prevent this projection-induced variability: frontal, lateral, top, and the plane defined by the first two principal axes of the cluster (Fig. 4, b and c). The value of K will be the maximum obtained in these four planes. The principal axes are the directions where the variance of data is maximized and, therefore, where the points exhibit the largest range. The first two principal axes define the principal plane of the cluster and are obtained by applying singular value decomposition (SVD) to the set of points forming the cluster.

Fig. 4. Should be placed here

Method 2

The second method applies a decision tree based on cluster features. The first step is to extract the following features for each cluster: volume (V), number of cluster points (P), mean reflectance of cluster points ($\bar{R}$), and a geometric parameter, Ψ, computed as the product of normalized eigenvalues $[\lambda_{1n}, \lambda_{2n}, \lambda_{3n}]$. The volume ($V$) was defined as the volume enclosed by the boundary points of the cluster. Clusters that contain more than one fruit are expected to have a larger volume (V) and more points (P). However, when a fruit is placed next to a leaf or trunk (not filtered in previous steps), the cluster volume (V) and the number of points (P) could increase as well. Due to this fact, the threshold, $\bar{R}$, is applied, as it was observed that the mean reflectance of this kind of trunk/leaf co-located cluster is lower than clusters containing grouped fruits. The last features used are the eigenvalues, which provide information about the cluster shape. Spherical shapes (clusters with only one apple) will have similar eigenvalues, while elongated shapes will have different eigenvalues. Eigenvalues are obtained with SVD, and their values depend on the variance of the points projected on the principal axes. In order to compare eigenvalues of different clusters, a normalization step is applied so that the eigenvalues sum to one. From that, the geometric parameter Ψ is defined as the product of eigenvalues and a normalization factor. The normalization factor of 27 allows the geometrical parameter, Ψ, to be bound between 0 and 1:

$$\lambda_{in} = \frac{\lambda_i}{\lambda_1+\lambda_2+\lambda_3} \quad \text{so that} \quad \lambda_{1n} + \lambda_{2n} + \lambda_{3n} = 1 \tag{1}$$

$$\Psi = 27 \cdot \lambda_{1n} \cdot \lambda_{2n} \cdot \lambda_{3n} \quad \text{where} \quad \begin{cases} \Psi = 1 & \text{for spherical distributions} \\ 1 > \Psi \geq 0 & \text{otherwise} \end{cases} \tag{2}$$

Fig. 5. Should be placed here

The implemented decision tree is based on the analysed features in the training data set and is composed of the following steps (Fig. 5):

- Feature extraction: Compute V ,P ,$\bar{R}$, and Ψ of the studied cluster.
- Step 1: If V, P, and $\bar{R}$ are higher than the corresponding thresholds V_{k1} , P_{k1}, $\bar{R}_{k1}$, and Ψ is smaller than Ψ_{k1}, it is concluded that the cluster contains more than one apple. Otherwise, K is assigned the value *1*.
- Step 2: A cluster will have more than two apples if P is higher than P_{k2} and Ψ is lower than Ψ_{k2} , or if V is higher than V_{k2}. Otherwise, K is assigned the value of *2*.
- Step 3: $K=4$ when a cluster meets both previous conditions and has a volume (V) higher than V_{k3}. Otherwise, K is assigned the value *3*.

All threshold values used in the decision tree were empirically selected by the graphical representation of four analysed features using the training dataset. The values used and the graphical representation of these features are presented in Appendix A. , Table A1 and Fig. A1.

Method 3

By applying method 1, some single-fruit clusters are split into multiple detections due to partial occlusions of apples by leaves. Method 3 addresses this concern by combining methods 1 and 2. First, step 1 of method 2 is applied to distinguish between clusters with single or multiple apples. For those clusters that contain more than one apple, method 1 is applied to determine the value of K.

2.3.3. False Positive removal

After implementing the first two steps of the algorithm (segmentation and apple separation), it was observed that some detections do not actually correspond to apples, i.e., these were false positive detections. That is because some leaves and trunks have a texture or shape that result in a high reflectance. It was found that some of these erroneous detections had a

different geometric shape (Ψ), volume (V), and mean reflectance ($\bar{R}$) compared to the successful detections. In order to reduce these false positives, the clusters that met the condition $(\Psi < \Psi_{FP}) \mid (\bar{R} < \bar{R}_{FP}) \mid (V > V_{FP})$ were removed. In the same manner as with method 2, the thresholds were empirically selected from a graphical representation of these three features using the training dataset. The values used and the graphical representation of these features are presented in Appendix A. , Table A1 and Fig. A2.

2.4. Performance evaluation

In this work, the results were evaluated using two different approaches: localization and identification. The *localization evaluation* aims to assess the system in the context of harvesting automation. This approach assumes that a robotic arm, when it gets close to a group of apples, is able to separate different apples that have been detected within the same cluster, or to unify the multi-detections that correspond to the same apple. Thus, a detection that contains K apples counts as K true positives (Fig. 6.a.), while multi-detections are counted as one true positive and no false positives (Fig. 6.e.).

The *identification evaluation* aims to assess the system for use in yield prediction or mapping. This assessment is performed cluster-by-cluster, so that a single detection containing K apples counts as only one true positive (Fig. 6.a.), while a single apple detected n times (multi-detection) is counted as one true positive and $N_m = n\text{-}1$ false positives (Fig. 6.e.).

To evaluate object detection in images, the metric intersection over union (IoU) is commonly used. This is possible when both bounding-box and object detection can be seen as a group of pixels. In this study, the detections are groups of 3D points, while ground truth bounding boxes are cube regions. The metric IoU has been substituted by the intersection over detection (IoD) for this reason; IoD is defined as the percentage of detected points that are placed inside ground truth bounding boxes.

The following defines the metrics used for each approach, namely localization (subscript L) and identification (subscript ID).

- Intersection over detection (IoD_i): Percentage of points, P_i, of a detection, i, that are placed inside ground truth bounding-boxes (GT). $IoD_i = \frac{P_i \cap GT}{P_i}$
- True positive localization (TP_L): Number of ground truth apples that are detected with an $IoD_i \geq 0.5$.
- False positive localization (FP_L): Number of detections with an $IoD_i < 0.5$.

- Localization success ($Success_L$): Quotient between TP_L and the number of labelled apples (GT_{labels}). $Success_L = \frac{TP_L}{GT_{labels}}$
- False detection rate localization (FDR_L): Ratio between FP_L and the total positive ($TP_L + FP_L$) $FDR_L = \frac{FP_L}{TP_L+FP_L}$
- True positive identification (TP_{ID}): Number of clusters with an $IoD_i \geq 0.5$, minus multi-detections ($\sum N_m$).
- False positive identification (FP_{ID}): Sum of the number of detections with an $IoD_i < 0.5$ (FP_L), plus multi-detections. $FP_{ID} = FP_L + \sum N_m$.
- Identification success ($Success_{ID}$ or recall): Quotient between TP_{ID} and GT_{labels}. $Success_{ID} = recall = \frac{TP_{ID}}{GT_{labels}}$
- False detection rate identification (FDR_{ID}): Ratio between FP_{ID} and the total positive. $FDR_{ID} = \frac{FP_{ID}}{TP_{ID}+FP_{ID}}$
- Precision: Percentage of TP_{ID} with respect to the total positive ($TP_{ID} + FP_{ID}$) $Precision = \frac{TP_{ID}}{TP_{ID}+FP_{ID}}$
- F1-score: Harmonic mean of precision and recall. $F1 = 2 \cdot \frac{precision \cdot recall}{precision+recall}$

Selected examples of the evaluation criteria can be seen in **Fig. 6**. Intersection over detection (IoD) is given for different scenarios, while true positive and false positive rates are calculated for localization and identification assessment approaches. Red shapes are apple detections, while green squares correspond to the ground truth labels. Note that actual clusters and bounding-boxes are in 3D (as shown in **Fig. 2**), although for the sake of simplicity this figure shows the 2D projection. The examples shown are: a) One cluster with K=2 apples and three GT bounding-boxes ; b) One cluster with K=1 apple and one GT bounding-box ; c) Two clusters of K=1 apple each and two GT bounding-boxes ; d) One GT bounding-box not detected ; e) Two clusters detecting the same GT bounding-box (multi-detection) ; f) One cluster that does not correspond to any GT object ; and g) One cluster detecting an apple with an IoD<0.5.

Fig. 6. Should be placed here

3. Results and discussion

3.1. Reflectance analysis

Table 1 shows the reflectance analysis results for both trees used in the training dataset. Mean apparent reflectance values of 28.9%, 29.1%, and 44.3% were obtained for leaves, trunks, and Fuji apples, respectively. These results indicate that the reflectance is higher than other tree elements. Hence, this characteristic will be used as a valuable feature for Fuji apple detection. Note that these results were obtained using a LiDAR system operating with a laser source at 905nm

wavelength. Further studies should be carried out to ensure that the present methodology could be extended to other laser systems (operating at different wavelengths) and other fruit varieties or branching structures.

Table 1. Reflectance analysis: The mean apparent reflectance and standard deviation of different elements in an apple orchard.

Tree	Elements	mean(R) [%]	std(R) [%]
T1	Leaves	29.23	13.57
T2	Leaves	28.69	13.88
T1	Trunks	29.67	14.83
T2	Trunks	28.52	15.41
T1	Apples	43.59	16.81
T2	Apples	45.10	16.78

The results of this analysis are the basis of the proposed detection algorithm, with reflectance being the principal feature used in the segmentation step. Although Fuji apples have higher reflectance than leaves and trunks at 905 nm, the standard deviation is high enough to create overlap between classes (**Fig. 7**.b). In order to find the optimal threshold, R_{th}, that will remove the points corresponding to leaves, branches, and trunks, a performance evaluation of the detection algorithm (Section 2.3) was carried out using different reflectance thresholds. **Fig. 7**.a plots the evolution of precision, recall, and F1-score metrics, computed before applying the false positive removal step, under different reflectance thresholds, $R_{th.}$. The best results were obtained with an $R_{th} = 60\%$, resulting in an *F1-score=82.16%* for the training dataset. **Fig. 7**.b shows the reflectance distributions for leaves, trunks, and apples. As can be seen, most of the 3D points belonging to apples were below the threshold value. This is because our restrictive threshold minimizes the false positives of leaves and trunks being selected as apples. Furthermore, omitting points as apple is not as critical as having a few apple points in a cluster, which are sufficient for detection.

Fig. 7. Should be placed here.

3.2. Step-by-step algorithm performance evaluation

This section presents a qualitative and quantitative evaluation of the different steps and methods implemented in this paper. Regarding the qualitative evaluation, **Fig. 8** illustrates the evolution after each processing step. First, **Fig. 8**a shows an RGB image of one of the trees, which is incorporated to assist in visualization, but was not used in the algorithm. **Fig. 8**b renders the 3D model obtained with the MTLS. The colour scale indicates the reflectance of each point, where blue

corresponds to low values and red implies high reflectance. It is evident from this representation how Fuji apples exhibit higher reflectance than other tree elements. **Fig. 8**c shows the results after applying the reflectance threshold, R_{th}, to the original point cloud. In this step, many of the leaf and trunk points were removed. Once the sparse outlier removal is applied (**Fig. 8**d), zones with low point density were removed, leaving only groups of points which are candidates for apple detection. **Fig. 8**e illustrates the segmentation output, which terminates the clustering of connected points. This result has clusters with one apple (red, orange, blue, and purple), clusters with more than one apple (green), and false positives (grey). The apple detection algorithm ends by splitting clusters with more than one apple and removing false positives. The final result is presented in **Fig. 8**f.

Table 2 presents the results of the test dataset for each step and method implemented. The first row shows the results after point cloud segmentation (Section 2.3.1); rows 2-4 indicate the results obtained when applying the splitting techniques presented in Section 2.3.2; and the last three rows present the final results after removing the false positives detected (Section 2.3.3).

Fig. 8. Should be placed here.

The localization success values obtained after point cloud segmentation (before apple separation and false positive removal) are slightly higher than 87% (first row). These results are similar to other methodologies using colour cameras (Gongal *et al*., 2015). The identification success presents significantly lower results (~73%), because of some detections containing more than one apple. Methods 1, 2, and 3 therefore were applied, in order to split these clusters, methods (Section 2.3.2). As a result, the identification success increased by more than 8% (rows 2 to 4), although the number of false positives also increased due to multi-detections. Method 1 performed best in terms of increasing the identification success (+11%), but also generated more multi-detections. Method 2 increased identification success by more than 8%, while false positives only increased 3%. The results of method 3 are a trade-off between the previous two methods. Since localization success performs an evaluation on a point-by-point basis, applying separation methods does not vary the results of this metric.

When applying false positive removal (rows 5 to 7), it is observed that the false detection rate fell by more than 5%, while the localization and identification successes were not affected (except for method 3, with a decrease of less than 1%). The best results were obtained by combining method 2 with false positive removal, resulting in a lower number of false positives, without affecting the performance of the apple detection algorithm.

The processing times indicated in Table 2 correspond to processing the data with a 64-bit operating system, with 8GB of RAM and an Intel ® Core(TM) i7-4500U processor (1.80 GHz, boosted to 2.40 GHz). Although method 2 was slightly more efficient than the other two approaches, no significant differences were observed in the processing time. This is because the most computationally intensive operation is in the DBSCAN clustering algorithm (9.1 seconds), which is part of the segmentation step included in all methods.

Table 2. Performance assessment of the different implemented steps and methods: point cloud segmentation (S); apple separation methods 1, 2, and 3 (M1, M2, and M3, respectively); and false positive removal step (FPr). Results include information from the test dataset (tree 3).

	Localization		Identification		
Method	$Success_L$ [%]	FDR_L [%]	$Success_{ID}$ [%]	FDR_{ID} [%]	Processing Time [s]
S	87.5	11.9	73.5	13.8	11.0
S + M1	87.5	20.7	85.3	26.1	11.9
S + M2	87.5	15.6	82.4	17.0	11.0
S + M3	87.5	16.8	84.6	20.7	12.0
S + M1 + FPr	87.5	12.5	85.3	18.3	12.1
S + M2 + FPr	87.5	9.8	82.4	10.4	11.1
S + M3 + FPr	86.8	11.3	83.8	14.9	12.4

3.3. Detection results

Table 3 shows the apple detection algorithm, as evaluated individually for each tree. These results were generated by applying the point cloud segmentation, followed by an apple separation using method 2, and removing false positives using the condition expressed in Section 2.3. The detection rate is similar for processed trees despite being slightly better for tree 1 and 3. A localization success of 87.5% with a 9.8% of FDR_L, an identification success of 82.4% with a 10.4% of $FDR_{ID,}$ and an F1-score of 85.8% were obtained using the test dataset. These results are comparable with those obtained with other methodologies used in the state of the art. So far, the best detection rates have been reported with image processing, obtaining accuracies of between 80% and 85% using colour features (Gongal *et al*., 2015), and up to 86% of recall using deep learning (Bargoti and Underwood, 2017). However, the vision systems used in harvesting robots in orchard environments report a mean value of 80% in localization success and a mean value of 70% in identification success (Bac *et al*., 2014). Although it is difficult to compare the research found in the state of the art review, given that they are evaluated

with different datasets, the methodology presented in this paper yields similar detection rates to previous work based on colour cameras, with the advantage that LiDAR-based measurements are not affected by illumination conditions. Furthermore, the location of each detected apple is obtained directly, which makes the presented system very interesting for autonomous harvesting or fruit load assessment for yield mapping applications.

Table 3. Apple detection assessment using method 2. Trees 1 and 2 were used as training dataset and tree 3 as test dataset. GT_{field} corresponds to the number of apples hand-counted in field, while GT_{labels} corresponds to the number of apples labelled in data. Other metrics are defined in Section 2.4.

			Localization				Identification				
Tree	GT_{field}	GT_{labels}	TP_L	FP_L	$Success_L$	FDR_L	TP_{ID}	FP_{ID}	$Success_{ID}$	FDR_{ID}	F1-score
Tree 1	139	133	116	12	87.2%	9.4%	110	16	82.7%	12.7%	0.849
Tree 2	145	138	118	13	85.5%	9.9%	109	15	79.0%	12.1%	0.832
Tree 3	139	136	119	13	87.5%	9.8%	112	13	82.4%	10.4%	0.858

Regarding the computational cost, Table 4 includes the inference time, processing each tree separately, and all trees combined. The number of points of each test is also reported. As expected, the computational time increases with the number of points processed. Results show that processing trees individually is much more efficient than processing all trees at once. This is because the average run time complexity of DBSCAN is not linear with the number of points (Ester *et al*., 1996), resulting in higher efficiency when processing small point clouds.

Table 4. Computational cost according to the number of points in the point cloud.

Tree	Nº of points	Processing Time [s]
Tree 1	438.260	8.0
Tree 2	460.847	9.6
Tree 3	526.136	11.2
Tree 1+2+3	1.425.243	68.8

4. Conclusions

This work presents a new methodology for Fuji apple detection and localization in real commercial orchard environments using a LiDAR-based mobile terrestrial laser scanner (MTLS) with reflectance capabilities. A reflectance analysis of the different apple tree elements was carried out, which showed that apples exhibit a higher reflectance than leaves and trunks at the 905 nm laser wavelength; we therefore conclude that this characteristic is a valuable feature for apple detection. An apple detection algorithm, suitable for dealing with point clouds obtained with an MTLS, was

subsequently developed and tested on three apple trees from a commercial apple orchard. The algorithm is divided into three steps: (1) removal of points corresponding to leaves and trunk and clustering the remaining points with a connected component labelling, (2) identification and splitting of clusters that contain more than one apple, and (3) false positive reduction. In order to predict the number of apples grouped in a cluster, three different methods were proposed: template matching, decision tree, and a combination of both approaches. The best results were achieved by applying a decision tree, resulting in a localization success of 87.5% with a 9.8% false detection rate, an identification success of 82.4% with a 10.4% false detection rate, and an F1-score of 85.8% in the test dataset. These outcomes represent an advance in the fruit detection field, since the results are comparable with those from colour (RGB) camera systems used in past efforts; however, the proposed LiDAR-based has the additional advantages that measurements are not affected by illumination conditions and that the method directly provides 3D fruit location information. An important limitation of this work is the small dataset. A larger dataset could allow the parameters to be learnt automatically (instead of being manually selected), thereby obtaining an algorithm that could better generalize with new data. Future efforts should include an analysis of fruit reflectance under different laser wavelengths, the extension of the dataset to other fruit varieties and species, and the application of machine learning algorithms in larger datasets.

Acknowledgements

This work was partly funded by the Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya (grant 2017 SGR 646), the Spanish Ministry of Economy and Competitiveness (projects AGL2013-48297-C2-2-R and MALEGRA, TEC2016-75976-R) and the Spanish Ministry of Science, Innovation and Universities (project RTI2018-094222-B-I00). The Spanish Ministry of Education is thanked for Mr. J. Gené's pre-doctoral fellowships (FPU15/03355). The work of Jordi Llorens was supported by Spanish Ministry of Economy, Industry and Competitiveness through a postdoctoral position named Juan de la Cierva Incorporación (JDCI-2016-29464_N18003). We would also like to thank CONICYT/FONDECYT for grant 1171431 and CONICYT FB0008. Nufri (especially Santiago Salamero and Oriol Morreres) and Vicens Maquinària Agrícola S.A. are also thanked for their support during the data acquisition.

Appendix A. Parameter values and feature analysis

Table A1 presents the values set for each parameter used in the algorithm. Parameter R_{th} is used in the segmentation step. See more details about these parameters in Section 2.3.1. Parameters with sub-index *kj* refer to the thresholds used in Section 2.3.2 - method 2 and were selected after analysing the graphical representation of cluster features shown in Fig. A1. Parameters with sub-index *FP* correspond to the thresholds used to remove false positives (Section 2.3.3) and were selected after analysing the graphical representation of detection features shown in Fig. A2.

Table A1. Parameter values used to detect apples in the presented dataset. The first five parameters were used during the point cloud segmentation step. Parameters sub-indexed with letter *K* correspond to thresholds used in the apple separation step. Parameters sub-indexed with letters *FP* were used in the false positive removal step.

Symbol	Value	Units
R_{th}	60	%
k	20	Points
α	0	---
$minPts$	15	Points
ε	0.03	m
V_{k1}	1.5·10-4	m^3
P_{k1}	85	Points
$\bar{R}_{k1}$	67.5	%
Ψ_{k1}	0.8	---
P_{k2}	400	Points
Ψ_{k2}	0.6	---
V_{k2}	1.2·10-3	m^3
V_{k3}	1.6·10-3	m^3
Ψ_{FP}	0.46	---
$\bar{R}_{FP}$	65.25	%
V_{FP}	10^{-3}	m^3

Fig. A1. Should be placed here.

Fig. A2. Should be placed here.

Table Captions

Table 5. Reflectance analysis. Mean reflectance and standard deviation.

Table 6. Performance assessment of the different implemented steps and methods: point cloud segmentation (S); apple separation methods 1, 2, and 3 (M1, M2, and M3, respectively); and false positive removal step (FPr). Results include information from the test dataset (tree 3).

Table 7. Apple detection assessment using method 2. Trees 1 and 2 were used as training dataset and tree 3 as test dataset. GT_{field} corresponds to the number of apples hand-counted in field, while GT_{labels} corresponds to the number of apples labelled in data. Other metrics are defined in Section 2.4.

Table 8. Computational cost according to the number of points in the point cloud.

Table A2. Parameter values

Figure Captions

Fig. 9. View of the MTLS equipment showing the GNSS antenna placement and the mounting orientation of the LiDAR sensor. Distance data are in mm.

Fig. 10. 3D point cloud models obtained for trees 1, 2, and 3. First two trees were used as training dataset, while the third one was kept as test dataset. Ground truth bounding boxes of tree 3 are shown, while the zoom bounding box (red circle) shows its shape.

Fig. 11. Apple detection algorithm flowchart.

Fig. 12. Method 1 - Cluster splitting by Gaussian smoothing. The aim of this method is to determine the number of apples, K, that are contained in a cluster. a) Actual data before applying method 1: the real scene is scanned and the resulting point cloud is segmented, obtaining clusters likely to contain apples. b) Cluster containing 2 apples. c) 2D projection in four planes: (1) frontal; (2) lateral, (3) top; and (4) plane defined by 2 principal axes. d) Gaussian smoothing. e) Local maxima identification.

Fig. 13. Method 2 - Decision tree used to predict the number of apples in a cluster.

Fig. 14. Localization and identification performance evaluation criteria. Intersection over detection (IoD) is given for different scenarios, while true positive and false positive are calculated for localization and identification assessment approaches. Red shapes are apple detections, while green squares correspond to the ground truth labels.

Fig. 15. a) Precision (green dashed line), recall (red dotted line), and F1-score (blue solid line) versus the applied reflectance threshold; b) Gaussian distributions obtained for each tree element in the reflectance analysis of the training dataset. Green solid line corresponds to leaves, blue dotted line to trunks and red dashed line to fruits. The vertical dash-dotted line indicates the reflectance threshold used for fruit detection.

Fig. 16. Illustration of the different processing steps (tree 2). a) RGB image. b) Point cloud obtained with the MTLS. c) Point cloud after applying the reflectance threshold. d) Sparse outlier removal. e) Connected component labelling (DBSCAN). f) Apple separation and false positive removal. For better visualization purposes, in b), scale ranges from 0 (blue) to 100 (red), while in c) and d) the scale ranges from 60 (blue) to 100 (red).

Fig. A3. Graphical representation of cluster features. The features analysed are the geometric parameter, Ψ, and the number of points (left), and the mean reflectance and the cluster volume (right). Clusters with one apple are represented in green squares; clusters with two apples are represented in blue diamonds; clusters with three apples in magenta asterisks; and clusters with four apples or more in black crosses. Yellow, red and blue lines correspond to K1, K2 and K3 thresholds, respectively. This analysis was performed on the training data set (Trees 1 and 2) and was used to set the thresholds explained in Section 2.3.2 - method 2.

Fig. A4. Graphical representation of detection features. The features analysed are the geometric parameter, Ψ, the mean reflectance and the cluster volume. False positives (FP) are represented by red crosses; true positives are represented by blue circles. Horizontal and vertical lines show the thresholds used to remove FP. This analysis was performed on the training data set (Trees 1 and 2) and was used to set the thresholds explained in Section 2.3.3.

REFERENCES

Auat Cheein, F., Torres-Torriti, M., Hopfenblatt, N.B., Prado, Á.J., Calabi, D., 2017. Agricultural service unit motion planning under harvesting scheduling and terrain constraints. J. F. Robot. 34, 1531–1542. doi:10.1002/rob.21738

Auat Cheein, F.A., Carelli, R., 2013. Agricultural robotics: Unmanned robotic service units in agricultural tasks. IEEE Ind. Electron. Mag. 7, 48–58. doi:10.1109/MIE.2013.2252957

Bac, C.W., Van Henten, E.J., Hemming, J., Edan, Y., 2014. Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead. J. F. Robot. 31. doi:10.1002/rob.21525

Bargoti, S., Underwood, J., 2017. Deep Fruit Detection in Orchards. 2017 IEEE Int. Conf. Robot. Autom. 3626–3633.

Barnea, E., Mairon, R., Ben-Shahar, O., 2016. Colour-agnostic shape-based 3D fruit detection for crop harvesting robots. Biosyst. Eng. 146, 57–70. doi:10.1016/j.biosystemseng.2016.01.013

Bechar, A., Vigneault, C., 2017. Agricultural robots for field operations. Part 2: Operations and systems. Biosyst. Eng. 153, 110–128.

doi:10.1016/j.biosystemseng.2016.11.004

Bechar, A., Vigneault, C., 2016. Agricultural robots for field operations: Concepts and components. Biosyst. Eng. 149, 94–111. doi:10.1016/j.biosystemseng.2016.06.014

Brunelli, R., 2009. Template Matching Techniques in Computer Vision - Theory And Practice. John Wiley & Sons.

Bulanon, D.M., Burks, T.F., Alchanatis, V., 2009. Image fusion of visible and thermal images for fruit detection. Biosyst. Eng. 103, 12–22. doi:10.1016/j.biosystemseng.2009.02.009

Bulanon, D.M., Burks, T.F., Alchanatis, V., 2008. Study on temporal variation in citrus canopy using thermal imaging for citrus fruit detection. Biosyst. Eng. 101, 161–171. doi:10.1016/j.biosystemseng.2008.08.002

Cariou, C., Lenain, R., Thuilot, B., Berducat, M., 2009. Automatic guidance of a four-wheel-steering mobile robot for accurate field operations. J. F. Robot. doi:10.1002/rob.20282

Chaivivatrakul, S., Dailey, M.N., 2014. Texture-based fruit detection. Precis. Agric. 15, 662–683. doi:10.1007/s11119-014-9361-x

De-An, Z., Jidong, L., Wei, J., Ying, Z., Yu, C., 2011. Design and control of an apple harvesting robot. Biosyst. Eng. 110, 112–122. doi:10.1016/j.biosystemseng.2011.07.005

Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proc. 2nd Int. Conf. Knowl. Discov. Data Min. doi:10.1.1.71.1980

Foglia, M.M., Reina, G., 2006. Agricultural robot for radicchio harvesting. J. F. Robot. doi:10.1002/rob.20131

Font, D., Pallejà, T., Tresanchez, M., Runcan, D., Moreno, J., Martínez, D., Teixidó, M., Palacín, J., 2014. A proposal for automatic fruit harvesting by combining a low cost stereovision camera and a robotic arm. Sensors (Switzerland) 14, 11557–11579. doi:10.3390/s140711557

Gong, A., Yu, J., He, Y., Qiu, Z., 2013. Erratum to " Citrus yield estimation based on images processed by an Android mobile phone ." Biosyst. Eng. 116, 111–112. doi:10.1016/j.biosystemseng.2013.07.004

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., Lewis, K., 2015. Sensors and systems for fruit detection and localization: A review. Comput. Electron. Agric. 116, 8–19. doi:10.1016/j.compag.2015.05.021

Gongal, A., Silwal, A., Amatya, S., Karkee, M., Zhang, Q., Lewis, K., 2016. Apple crop-load estimation with over-the-row machine vision system. Comput. Electron. Agric. 120, 26–35. doi:10.1016/j.compag.2015.10.022

Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern Recognit. Lett. doi:10.1016/j.patrec.2009.09.011

Jiménez, A.R., Ceres, R., Pons, J.L., 2000. A vision system based on a laser range-finder applied to robotic fruit harvesting. Mach. Vis. Appl. 11, 321–329. doi:10.1007/s001380050117

Jiménez, A.R., Jain, A.K., Ceres, R., Pons, J.L., 1999. Automatic fruit recognition: a survey and new results using Range/Attenuation images. Pattern Recognit. 32, 1719–1736. doi:10.1016/S0031-3203(98)00170-8

Kaasalainen, S., Jaakkola, A., Kaasalainen, M., Krooks, A., Kukko, A., 2011. Analysis of incidence angle and distance effects on terrestrial laser scanner intensity: Search for correction methods. Remote Sens. doi:10.3390/rs3102207

Kukko, A., Kaasalainen, S., Litkey, P., 2008. Effect of incidence angle on laser scanner intensity and surface data. Appl. Opt. doi:10.1364/ao.47.000986

Kurtulmus, F., Lee, W.S., Vardar, A., 2014. Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network. Precis. Agric. 15, 57–79. doi:10.1007/s11119-013-9323-8

Lak, M.B., Minaei, S., Amiriparian, J., Beheshti, B., 2010. Apple fruits recognition under natural luminance using machine vision. Adv. J. Food Sci. Technol. 2, 325–327.

Linker, R., 2017. A procedure for estimating the number of green mature apples in night-time orchard images using light distribution and its application to yield estimation. Precis. Agric. 18, 59–75. doi:10.1007/s11119-016-9467-4

Linker, R., Cohen, O., Naor, A., 2012. Determination of the number of green apples in RGB images recorded in orchards. Comput. Electron. Agric. 81, 45–57. doi:10.1016/j.compag.2011.11.007

Liu, X., Zhao, D., Jia, W., Ruan, C., Tang, S., Shen, T., 2016. A method of segmenting apples at night based on color and position information. Comput. Electron. Agric. 122, 118–123. doi:10.1016/j.compag.2016.01.023

Maldonado, W., Barbosa, J.C., 2016. Automatic green fruit counting in orange trees using digital images. Comput. Electron. Agric. 127, 572–581. doi:10.1016/j.compag.2016.07.023

Nguyen, T.T., Vandevoorde, K., Wouters, N., Kayacan, E., De Baerdemaeker, J.G., Saeys, W., 2016. Detection of red and bicoloured apples on tree with an RGB-D camera. Biosyst. Eng. 146, 33–44. doi:10.1016/j.biosystemseng.2016.01.007

Okamoto, H., Lee, W.S., 2009. Green citrus detection using hyperspectral imaging. Comput. Electron. Agric. 66, 201–208. doi:10.1016/j.compag.2009.02.004

Payne, A., Walsh, K., Subedi, P., Jarvis, D., 2014. Estimating mango crop yield using image analysis using fruit at “stone hardening” stage and night time imaging. Comput. Electron. Agric. 100, 160–167. doi:10.1016/j.compag.2013.11.011

Qureshi, W.S., Payne, A., Walsh, K.B., Linker, R., Cohen, O., Dailey, M.N., 2017. Machine vision for counting fruit on mango tree canopies. Precis. Agric. 18, 224–244. doi:10.1007/s11119-016-9458-5

Ray, T.W., 1994. A FAQ on vegetation in remote sensing. California: Div. of Geological and Planetary Sciences California Institute of Technology.

Rosell-Polo, J.R., Cheein, F.A., Gregorio, E., Andújar, D., Puigdomènech, L., Masip, J., Escolà, A., 2015. Advances in Structured Light Sensors Applications in Precision Agriculture and Livestock Farming, in: Advances in Agronomy. doi:10.1016/bs.agron.2015.05.002

Rosell-Polo, J.R., Gregorio, E., Gene, J., Llorens, J., Torrent, X., Arno, J., Escola, A., 2017. Kinect v2 Sensor-based Mobile Terrestrial Laser Scanner for Agricultural Outdoor Applications. IEEE/ASME Trans. Mechatronics 1–1. doi:10.1109/TMECH.2017.2663436

Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M., 2008. Towards 3D Point cloud based object maps for household environments. Rob. Auton. Syst. 56, 927–941. doi:10.1016/j.robot.2008.08.005

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., McCool, C., 2016. DeepFruits: A Fruit Detection System Using Deep Neural Networks. Sensors 16, 1222. doi:10.3390/s16081222

Safren, O., Alchanatis, V., Ostrovsky, V., Levi, O., 2007. Detection of Green Apples in Hyperspectral Images of Apple-Tree Foliage Using machine Vision 50, 2303–2313.

Si, Y., Liu, G., Feng, J., 2015. Location of apples in trees using stereoscopic vision. Comput. Electron. Agric. 112, 68–74. doi:10.1016/j.compag.2015.01.010

Siegel, K.R., Ali, M.K., Srinivasiah, A., Nugent, R.A., Narayan, K.M.V., 2014. Do we produce enough fruits and vegetables to meet global health need? PLoS One 9. doi:10.1371/journal.pone.0104059

Stajnko, D., Lakota, M., Hocevar, M., 2004. Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal

imaging. Comput. Electron. Agric. 42, 31–42. doi:10.1016/S0168-1699(03)00086-3

Stein, M., Bargoti, S., Underwood, J., 2016. Image Based Mango Fruit Detection, Localisation and Yield Estimation Using Multiple View Geometry. Sensors 16, 1915. doi:10.3390/s16111915

Velodyne, L., 2016. VLP-16 In VLP-16 Manual: User's Manual and Programming Guide; Velodyne LiDAR.

Wachs, J.P., Stern, H.I., Burks, T., Alchanatis, V., 2010. Low and high-level visual feature-based apple detection from multi-modal images. Precis. Agric. 11, 717–735. doi:10.1007/s11119-010-9198-x

Wehr, A., Lohr, U., 1999. Airborne laser scanning - An introduction and overview. ISPRS J. Photogramm. Remote Sens. doi:10.1016/S0924-2716(99)00011-8

Xiang, R., Jiang, H., Ying, Y., 2014. Recognition of clustered tomatoes based on binocular stereo vision. Comput. Electron. Agric. 106, 75–90. doi:10.1016/j.compag.2014.05.006

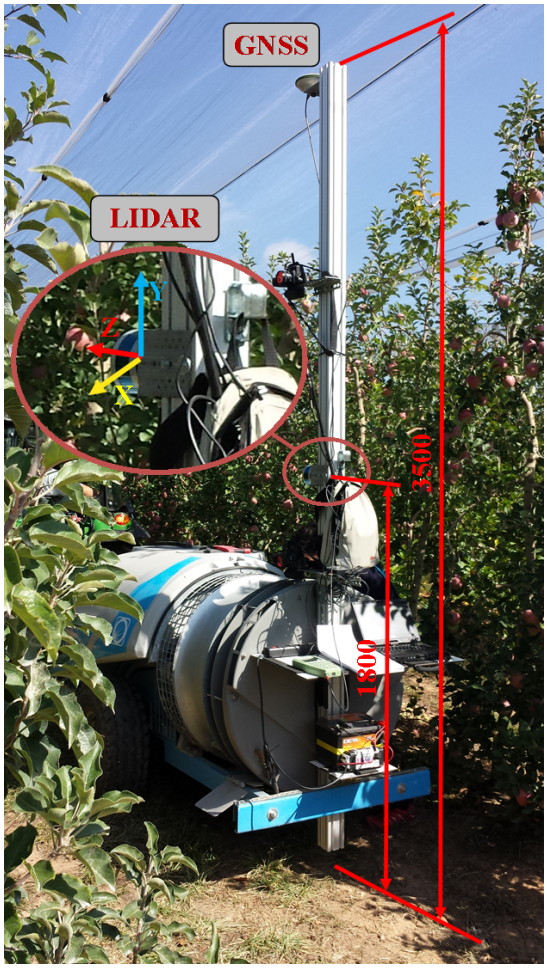
Zhang, B., Huang, W., Wang, C., Gong, L., Zhao, C., Liu, C., Huang, D., 2015. Computer vision recognition of stem and calyx in apples using near-infrared linear-array structured light and 3D reconstruction. Biosyst. Eng. 139, 25–34. doi:10.1016/j.biosystemseng.2015.07.011

Zhang, Q., Pierce, F.J., 2016. Agricultural automation: fundamentals and practices. CRC Press.

Zhao, C., Lee, W.S., He, D., 2016. Immature green citrus detection based on colour feature and sum of absolute transformed difference (SATD) using colour images in the citrus grove. Comput. Electron. Agric. 124, 243–253. doi:10.1016/j.compag.2016.04.009

Zhao, Y., Gong, L., Huang, Y., Liu, C., 2016. A review of key techniques of vision-based control for harvesting robot. Comput. Electron. Agric. 127, 311–323. doi:10.1016/j.compag.2016.06.022

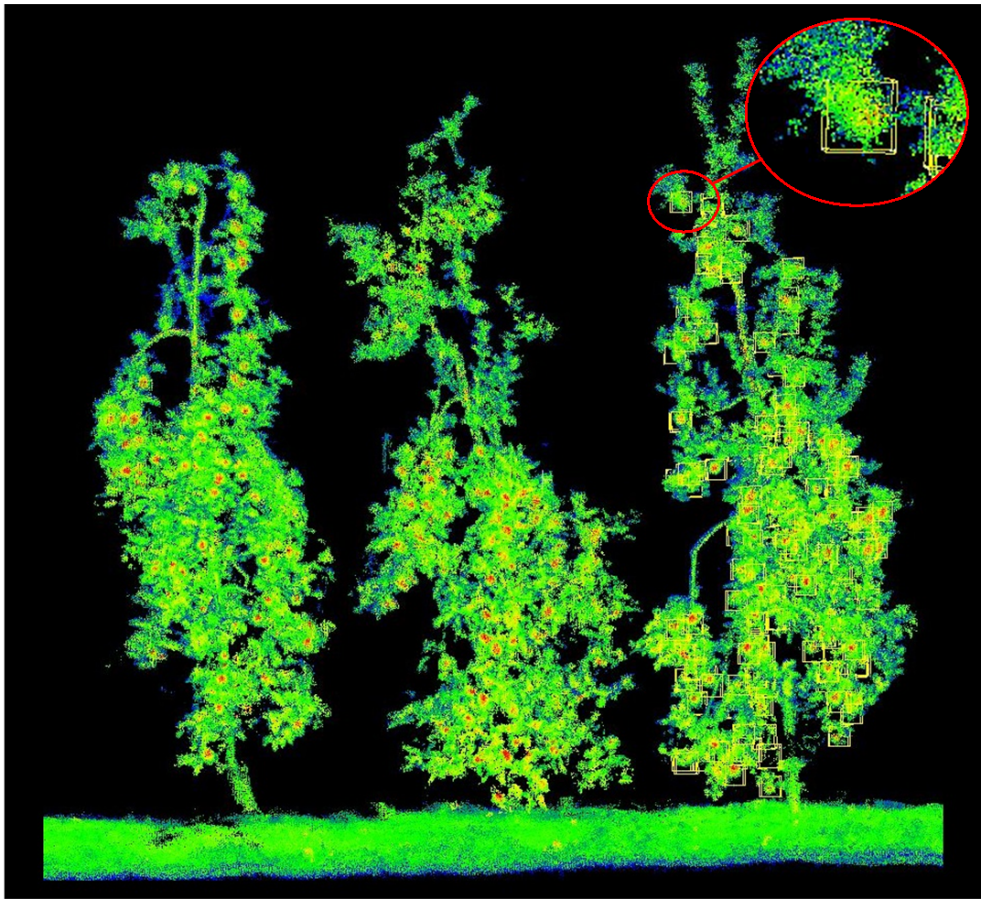
584 **Fig. 17.** View of the MTLS equipment showing the GNSS antenna placement and the mounting orientation of the LiDAR sensor.
585 Distance data are in mm.



586

587

588 **Fig. 18.** 3D point cloud models obtained for trees 1, 2, and 3. First two trees were used as training dataset, while the third was kept as test
589 dataset. Ground truth bounding boxes of tree 3 are shown, while the zoom bounding box (red circle) shows its shape.



590

591

Fig. 19. Apple detection algorithm flowchart.

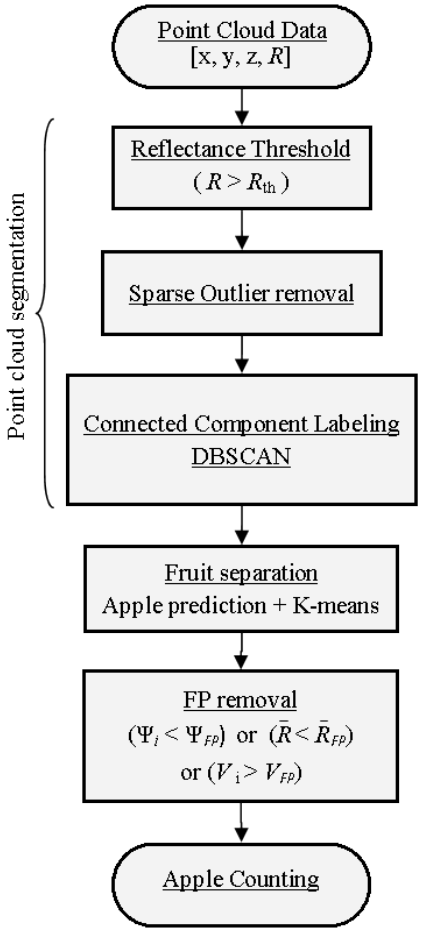


Fig. 20. Method 1 - Cluster splitting by Gaussian smoothing. The aim of this method is to determine the number of apples, K, that are contained in a cluster. a) Actual data before applying method 1: the real scene is scanned and the resulting point cloud is segmented, obtaining clusters likely to contain apples. b) Cluster containing 2 apples. c) 2D projection in four planes: (1) frontal; (2) lateral, (3) top; and (4) plane defined by 2 principal axes. d) Gaussian smoothing. e) Local maxima identification.

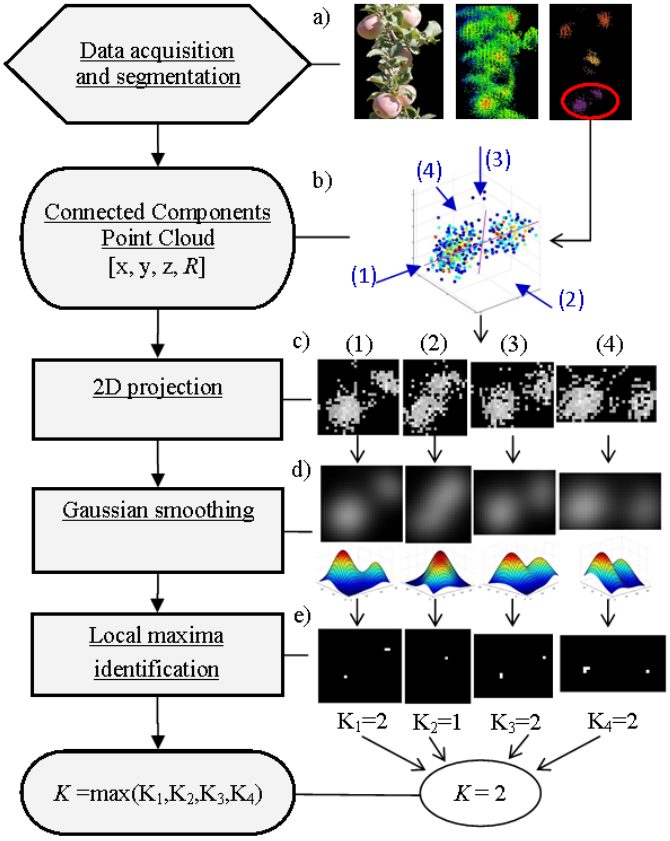


Fig. 21. Method 2 - Decision tree used to predict the number of apples in a cluster.

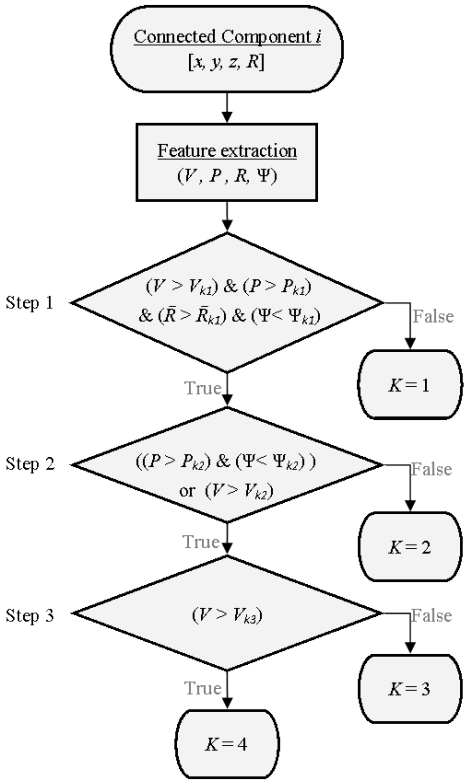
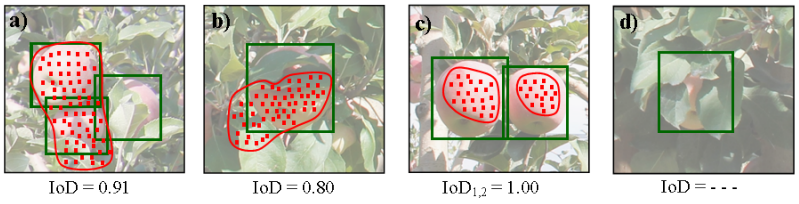
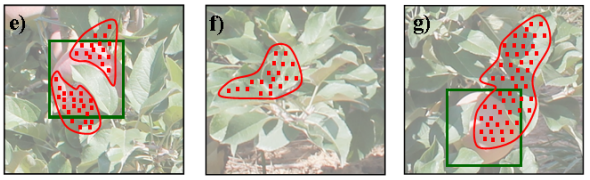


Fig. 22. Localization and identification performance evaluation criteria. Intersection over detection (IoD) is given for different scenarios, while true positive and false positive are calculated for localization and identification assessment approaches. Red shapes are apple detections, while green squares correspond to the ground truth labels.

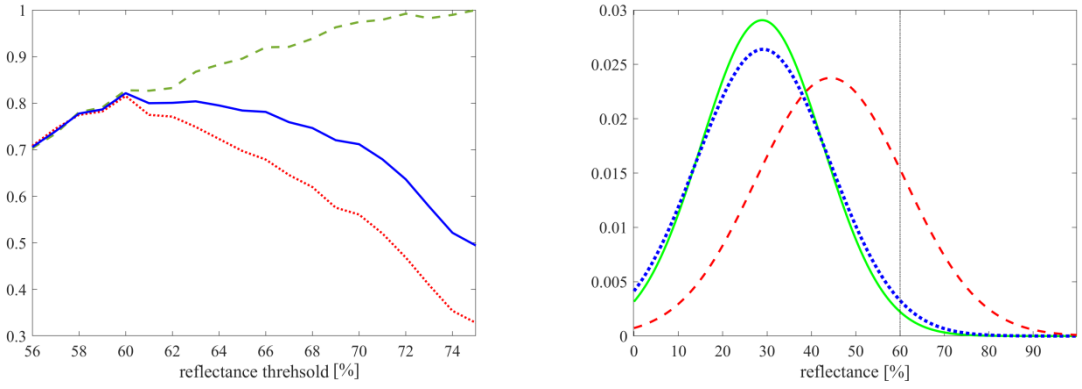


	a)	b)	c)	d)
	$IoD = 0.91$	$IoD = 0.80$	$IoD_{1,2} = 1.00$	$IoD = - - -$
Loc.	2 TP_L ; 0 FP_L	1 TP_L ; 0 FP_L	2 TP_L ; 0 FP_L	0 TP_L ; 0 FP_L
Id.	1 TP_{ID} ; 0 FP_{ID}	1 TP_{ID} ; 0 FP_{ID}	2 TP_{ID} ; 0 FP_{ID}	0 TP_{ID} ; 0 FP_{ID}



	e)	f)	g)
	$IoD_1 = 0.81$; $IoD_2 = 0.77$	$IoD = 0.00$	$IoD = 0.42$
Loc.	1 TP_L ; 0 FP_L	0 TP_L ; 1 FP_L	0 TP_L ; 1 FP_L
Id.	1 TP_{ID} ; 1 FP_{ID}	0 TP_{ID} ; 1 FP_{ID}	0 TP_{ID} ; 1 FP_{ID}

Fig. 23. a) Precision (green dashed line), recall (red dotted line), and F1-score (blue solid line) versus the applied reflectance threshold; b) Gaussian distributions obtained for each tree element in the reflectance analysis of the training dataset. Green solid line corresponds to leaves, blue dotted line to trunks and red dashed line to fruits. The vertical dash-dotted line indicates the reflectance threshold used for fruit detection.



a) **b)**

Fig. 24. Illustration of the different processing steps (tree 2). a) RGB image. b) Point cloud obtained with the MTLS. c) Point cloud after applying the reflectance threshold. d) Sparse outlier removal. e) Connected component labelling (DBSCAN). f) Apple separation and false positive removal. For better visualization purposes, in b), scale ranges from 0 (blue) to 100 (red), while in c) and d) the scale ranges from 60 (blue) to 100 (red).

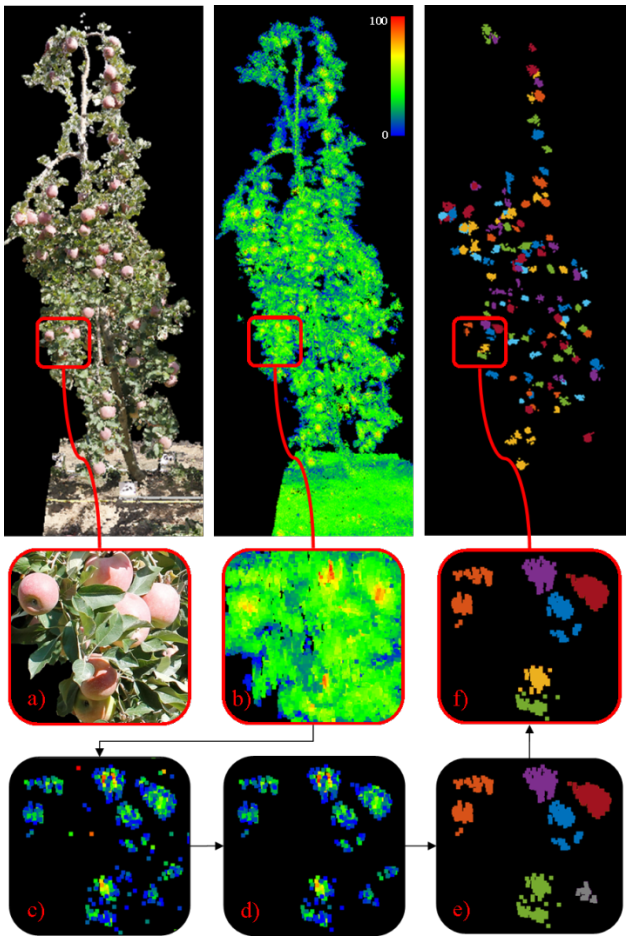


Fig. A5. Graphical representation of cluster features. The features analysed are the geometric parameter, Ψ, and the number of points (left), and the mean reflectance and the cluster volume (right). Clusters with one apple are represented in green squares; clusters with two apples are represented in blue diamonds; clusters with three apples in magenta asterisks; and clusters with four apples or more in black crosses. Yellow, red and blue lines correspond to K1, K2 and K3 thresholds, respectively. This analysis was performed on the training data set (Trees 1 and 2) and was used to set the thresholds explained in Section 2.3.2 - method 2.

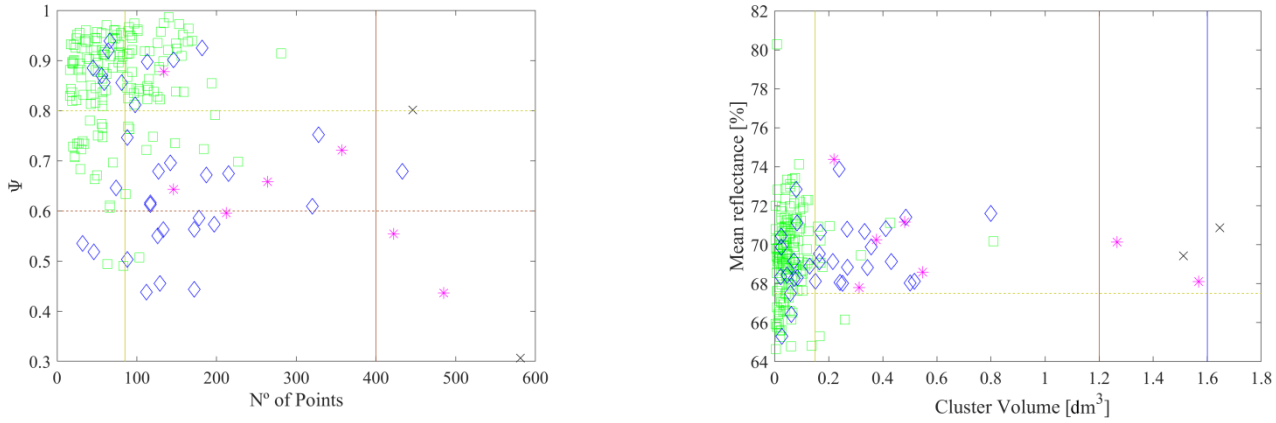


Fig. A6. Graphical representation of detection features. The features analysed are the geometric parameter, Ψ, the mean reflectance and the cluster volume. False positives (FP) are represented by red crosses; true positives are represented by blue circles. Horizontal and vertical lines show the thresholds used to remove FP. This analysis was performed on the training data set (Trees 1 and 2) and was used to set the thresholds explained in Section 2.3.3.